Hardspace, softspace and the possibilities of open source architecture

By Usman Haque, usman@haque.co.uk, www.haque.co.uk

Our constructed environment, with its direct impact on people every day and its constant transformation through use and reuse, is a collectively designed project. It incorporates vastly different and sometimes conflicting logics. The issues arising from people's differing perspectives and approaches will have significant consequences on the way architecture in general evolves in the twenty-first century. Computer terminology has borrowed much from the discipline of architecture; here, we borrow back some analogies from the computer world to suggest ways that architectural evolution could occur.

Traditionally, architecture has been thought of as hardware: the static walls, roofs and floors that enclose us. An alternative approach is to think of architecture as software: the dynamic and ephemeral sounds, smells, temperatures even radio waves that surround us. One might also consider the social infrastructures that underpin our designed spaces. Pushing this analogy even further, we can think of architecture as a whole as an "operating system", within which people create their own programmes for spatial interaction.

Architectural design that emphasises "softspace" over "hardspace" is a little like "software" design rather than "hardware design" in computer terminology, where "hardware" refers to the physical machine and "software" refers to the programs that animate the machine. In an architectural context, technology is used to provoke interactions between people, and between people and their spaces. If softspace encourages people to become performers within their own environments, then hardspace provides a framework to animate these interactions. The idea of an architectural operating system lies in the design of the systems that integrate the two. One model of operating system that is particularly relevant to architecture (since the design of space is always a collaborative process) is an open source system.

Everyone is a space designer and we all use our spaces and interfaces differently. We place posters on walls, paint them light blue or orange, position furniture in rooms, make love in kitchens, use "bedrooms" as "offices", sing opera in the shower, spray particular fragrances in our bathrooms and use staircases for arguments, games and romances. Meanwhile, we are increasingly likely to undertake the construction or improvement of our own homes without needing the services of an architect. Yet, most people do not think of themselves as being able to "design".

Even in architect-designed environments, technological developments throw into question the very role of the architect, because user- and environmentally-responsive mechanisms allow people themselves to take prime position in configuring (i.e. designing) their own spaces. The simplest form of such mechanisms is the thermostat, regulating temperature according to our requirements; more recently, systems that allow for changing colour, texture, layout and transparency of walls and other systems that suggest the circular process of "conversation" with one's environment have made it clear that architects no longer have priority in defining the boundaries of people's movements and desires. So what then does an architect do?

If an architect designs interaction systems then the production of architecture (which exists only at the moment of use) is placed in the hands of the end user. Architectural design, the choreography of sensations, can provide meta-programs within which people construct their own programs. In computers, an "operating system" is the software (like Unix, Windows NT or Mac OS X) that runs a computer at its core level and which provides a platform upon which to run other programs. Extending the analogy to architecture, a spatial operating system provides frameworks to encourage multitudes of architectural programs. In this conception, people are the designers of their own spaces — architects simply design the meta-systems.

Interaction systems conflate distinctions between audiences and performers; users and designers; occupants and architects and open up creative possibilities for designed space, designed events and designed situations. They also raise challenges for the social role of designers in providing meta-systems that foster individual creativity and encourage people to choreograph their own spatial programs, design their

own spaces and invent their own logics. <mark>The quandary is to design operating systems that promote creativity without adding further layers of prescriptive control.</mark>

"Open source" in the software universe refers to a type of source code (with which software is designed and built) that is accessible to all; that is freely distributed as long as it remains equally open; that allows for modification and derivatives as long as the result is equally open; that is non-discriminatory; where patching is possible without disturbing the integrity of the main work; and that is technology neutral.

Similarly, an open source architecture requires a framework in which the distinction between "those who design" and "those who use" is replaced by participatory system that encourages a constructed project to be constantly "patched" or "performed". Such an architecture comes close to the visions Dutch artist, architect and situationist Constant had in his project New Babylon. In this massive exploration he assumed that everyone is an artist in the design and construction of their spaces, events and lives. His project proposed a worldwide structure constantly built and rebuilt by its inhabitants, a structure that varied throughout its lengths as different groups of people contributed to it and altered it in different ways. He diminished the gaps between the practice of art and the practice of architecture and highlighted the connections between the delight of art and the delight of architecture. However, the project raises an important question: if everyone is an artist, and everything is art (read architecture) then does that not mean that, with no distinctions, nothing is art?

The open source movement in software gives us clues on how to resolve this conundrum – it offers a system that is in theory open enough that anyone might jump in but in practice has not meant that everyone is becoming a programmer. There are still those who enjoy the system for the challenge of building new code, and those who enjoy open source culture without needing to contribute to the construction process. Similarly, applying open source to architecture suggests a collaborative democratic project that exists in time as well as space: an architecture that is created by people through its use, as a performance, a conversation, a bodystorm that goes on throughout the life of the architectural system, whether it is a building or other architectural situation.

Industrial design theorist Anthony Dunne, talking about the design of what he calls "post-optimal objects" (i.e. objects one designs once practicality and functionality can be taken for granted) says: "the most difficult challenges for designers of electronic objects now lie not in technical and semiotic functionality, where optimal levels of performance are already attainable, but in the realms of metaphysics, poetry and aesthetics where little research has been carried out" (Hertzian Tales). If we assume that such systems in architecture could deal with the practical and functional requirements of constructed spaces then the beauty in design comes from the poetries of those who use/implement/remake it. A system that allows people themselves to create their own spaces and collectively build a social space -- that would be more conceptually "open".

There are several key features to an open source architecture:

1. Designer-participants: where those who participate are also those who design the system.
2. A control system that one allows oneself to be part of in order to expand that structure: an example can be found in computer games that provide modules for end-users to code and create their own, sometimes startlingly different, versions of the game.
3. Choreographies for openness: group instructions that are interpreted and modified as necessary by participants, individually or collectively. To begin, established boundaries are required in order to foster creativity; this does not mean that they cannot be breached. They are placed as reference points, not to pre-define limits.
4. Re-appropriation: where existing spaces, objects or actions are both fuel and catalysts for further creativity
5. Capacity for sharing design problems: each person has different skills and often a problem requires a solution that can only be provided by another. A web-based example, lazyweb.org, shows how it is not

important for everyone to have the technical capabilities in order to have an open source model of production.

In the immediate future, open source architecture would require two distinct steps.

First would be to develop infrastructures that enable "non-professional" designers to participate more closely in design and construction process. In some senses, this is already occurring, as the self-build trend shows. However, "professional" architects can do much more to facilitate the transition. Pragmatically, they have the opportunity at this stage to participate in the conversations that take place with regard to enabling and encouraging good building design and collaborative practice. This can occur at the practical level of expanding participatory practices in the industry; however, it can also occur in theoretical discourse where the very ideas and concepts behind architecture need to be opened.

Second would be to apply knowledge of space design to the formulation of a framework within which other people can consciously design spaces. In this capacity, architects would encourage recognition of the distinction between "good" design and "bad" design, if that can be said to exist. Again, this step can be located prosaically within current industry practice; however, it is also necessary to expand theoretical discourse on how to "design design". A spatial operating system acknowledges that everyone is already a designer: it would be vital with this step to ensure that architects don't become just another meta-system that "objectively" controls the process from above. Rather than directing, they would need primarily to become enablers or co-operants.

The role of architecture undergoes considerable change because people themselves interpret, appropriate, design and reuse a space within their own frames of logic. A truly open source architecture does not exist without people to inhabit, occupy, perceive, interact or converse with it. The resulting spaces don't merely enable people to develop their own ways of responding, they are actually enriched by them doing so. As people become architects of their own spaces (through their use) or developers of their own interfaces, the words "architecture" and "interface" cease to be nouns: instead they become verbs. Such an architecture is explicitly dynamic, a shift that opens up a wealth of poetic possibilities for designers of "open source" space.

We know that architecture is political. And we know that people themselves make architecture by using it. The challenge now is to balance the differences in technical skill, technology access and self-sufficiency desire that different people have, in order to produce a viably democratic space (in all senses). Are all architectural systems meta-systems of control? Open source and similar collaborative design processes suggest that there are other ways forward.